

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2001-223750

(43)Date of publication of application : 17.08.2001

(51)Int.Cl. H04L 12/56
G06F 17/30
H04L 12/46
H04L 12/28

(21)Application number : 2000-378532

(71)Applicant : ASCEND COMMUNICATIONS INC

(22)Date of filing : 13.12.2000

(72)Inventor : HEBB ANDREW T
CHERIAN SANJAY G

(30)Priority

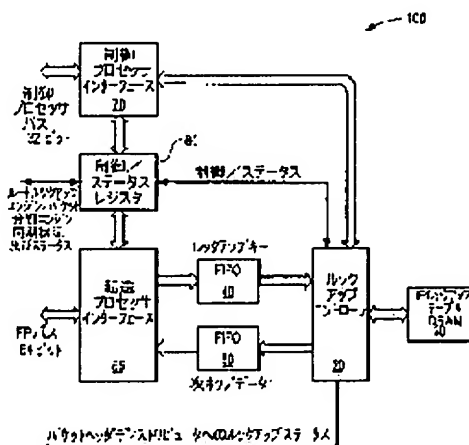
Priority number : 1999 459441 Priority date : 13.12.1999 Priority country : US

(54) ROUTE LOOKUP ENGINE

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a route lookup engine which speedily and effectively perform a route lookup process since a method by conventional technology used for matching in a conventional route lookup process is complicated and takes a long time.

SOLUTION: A route lookup engine(RLE) which determines a next hop index is disclosed. The RLE receives a lookup key and makes a multibit tree search by variable slide tree capture and variable slide tree capture. Data that the route lookup engine sends back consist of next hop information and a status flag. The RLE uses a compact and filed-reusable data structure. The RLE executes a unicast IP address lookup process and a multicast IP address lookup process over a virtual private network. The RLE uses different index memories and transfer memories. The upper limit of the search time of the RLE is determined irrelevantly to route table size.



LEGAL STATUS

[Date of request for examination]

31.05.2002

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2001-223750

(P2001-223750A)

(43) 公開日 平成13年 8月17日 (2001. 8. 17)

(51) Int.Cl. ⁷	識別記号	F I	テマコード [*] (参考)
H 0 4 L 12/56		G 0 6 F 17/30	1 7 0 Z
G 0 6 F 17/30	1 7 0		4 1 3
	4 1 3	H 0 4 L 11/20	1 0 2 Z
H 0 4 L 12/46		11/00	3 1 0 C
12/28			

審査請求 未請求 請求項の数21 O L (全 9 頁)

(21) 出願番号 特願2000-378532(P2000-378532)

(22) 出願日 平成12年12月13日 (2000. 12. 13)

(31) 優先権主張番号 0 9 / 4 5 9 4 4 1

(32) 優先日 平成11年12月13日 (1999. 12. 13)

(33) 優先権主張国 米国 (U S)

(71) 出願人 500570195

アセンド コミュニケーションズ インコ
ーポレーテッド

アメリカ合衆国 94502 カリフォルニア,
アラミーダ, ハーバー ベイ パークウェ
イ 1701, アセンド プラザ 1

(72) 発明者 アンドリュー テー. ヘブ

アメリカ合衆国 01749 マサチューセッ
ツ, ハドソン, レイクサイド アヴェニュー
ー 62

(74) 代理人 100064447

弁理士 岡部 正夫 (外11名)

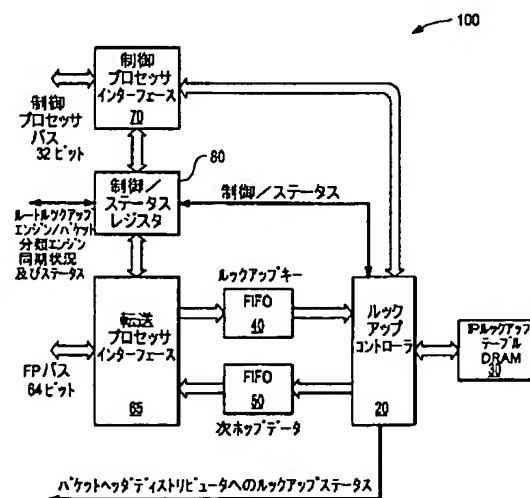
最終頁に続く

(54) 【発明の名称】 ルートルックアップエンジン

(57) 【要約】

【課題】 従来のルートルックアップにおいてマッチングを実行するために使用される従来技術による方法は複雑、かつ時間がかかる。よって、ルートルックアップを迅速、かつ効果的に実行するルートルックアップエンジンが必要とされる。

【解決手段】 本発明は次ホップインデックスを判断するルートルックアップエンジン (R L E) を開示する。R L Eはルックアップキーを受け取り、プレフィックスエクスパンションと可変ストライドツリーキャプチャによりマルチビット樹木探索を実行する。ルートルックアップエンジンが返送するデータは、次ホップ情報およびステータスフラグより構成される。R L Eはコンパクトでフィールド再利用可能なデータ構造を使用する。R L Eでは仮想プライベートネットワーク上にユニキャスト I Pアドレスルックアップとマルチキャスト I Pアドレスルックアップの両方を実行する。R L Eは別々のインデックスメモリーと転送メモリーを使用する。R L Eのサーチタイムの上界はルートテーブルサイズに関係なく決定される。



【特許請求の範囲】

【請求項1】 少なくとも1つのネットワークに対して少なくとも1つのツリーを含む1Pルックアップテーブルメモリと、

次ホップインデックスを得るため該1Pルックアップテーブルメモリとの通信を行い、そして該1Pルックアップテーブルメモリのサーチが可能なルックアップコントローラと、

1Pルックアップテーブルサーチ開始のために該ルックアップコントローラ用に該ルックアップコントローラに対しサーチキーを供給し、かつ、該ルックアップコントローラから該次ホップインデックスを受取る、該ルックアップコントローラと電気通信を行う転送プロセッサインターフェースとを備えていることを特徴とするルートルックアップエンジン。

【請求項2】 少なくとも1つのネットワークは仮想プライベートネットワークであることを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項3】 上記ルックアップコントローラと通信を行い、該1Pルックアップテーブルメモリについてのテーブル更新を行うプロセッサインターフェースを備えていることを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項4】 上記転送プロセッサインターフェースと上記ルックアップコントローラ間に配置されたルックアップキーメモリを備えていることを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項5】 ルックアップキーメモリはFIFO（先入れ先出し）メモリであることを特徴とする請求項4に記載のルートルックアップエンジン。

【請求項6】 上記ルックアップコントローラと上記転送プロセッサインターフェース間に配置された次ホップ結果メモリを備えていることを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項7】 次ホップ結果メモリはFIFOメモリであることを特徴とする請求項6に記載のルートルックアップエンジン。

【請求項8】 上記プロセッサインターフェース、上記転送プロセッサインターフェース、および上記ルックアップコントローラと通信を行う制御／ステータスレジスタを備えていることを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項9】 上記ルックアップコントローラはユニキャストルックアップを実行することを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項10】 上記ルックアップコントローラはマルチキャストルックアップを実行することを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項11】 上記サーチキーはIPアドレスにより構成されることを特徴とする請求項1に記載のルート

ルックアップエンジン。

【請求項12】 上記ルックアップメモリはDRAMから成ることを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項13】 上記少なくとも1つのツリーは可変ストライド（stride）ツリーから成ることを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項14】 上記ルックアップコントローラは、上記1Pルックアップテーブルメモリをサーチする際にブレフィックスエクスパンションおよびキャプチャを実行することを特徴とする請求項1に記載のルートルックアップエンジン。

【請求項15】 ルートルックアップリクエストを受取るステップと；ルートルックアップを実行するステップであって；そのルックアップが該リクエストに対する最初のルックアップであるかを判断するステップと、

それが該リクエストに対する最初のルックアップである時、該リクエストについて現ブロックデータを使用し、また、それが該リクエストに対する最初のルックアップでない時、前回のブロックデータを使用するステップと、

該ブロックデータが葉を示しているかどうかの判断を行うステップであって、

該ブロックデータが葉を示していない時、該ブロックデータからのルートルックアップエンジン根に加えて、該ブロックデータからの現オーダーデータを差し引いたソースアドレスの連続した宛先アドレスに連続するゼロビットから成るアドレスを含むルートテーブルルックアップアドレスを提供するステップと、

該ルックアップテーブルより読取りを行うステップであって、該読取りは新規ブロックデータを提供するステップと、

現時点の現オーダーデータ、プラス、新規ブロックデータからのオーダーデータ、に現オーダーデータを設定するステップと、

そのルックアップが該リクエストに対する最初のルックアップであるかどうかを判断するステップに戻るステップとから成るステップと；該データブロックが葉を示していない時、間接ビットが該ブロックデータに設定されているかを判断し、該間接ビットが設定がされている時、

該ブロックデータからのルートテーブルルックアップエンジンルートから成るルートテーブルルックアップアドレスを供給するステップと、

該ルックアップテーブルからの読取りを行い、該読取りは新規ブロックデータを供給するステップと、

該間接ビットが1に等しくないかどうか、またタイプビットがノードに等しくないかどうか結果を保存するこれらのステップを実行するステップと；該間接ビットが設定されていない時、結果を保存するステップを実行する

10

20

30

40

50

ステップとから成ることを特徴とするルートルックアップエンジンの動作方法。

【請求項16】 現オーダデータを設定するステップはさらに、上記の現オーダデータが64よりも小さいかあるいは64に等しい場合、エラーコンディションを示すステップから成ることを特徴とする請求項15に記載の方法。

【請求項17】 現オーダデータを設定するステップはさらに、上記の現オーダが32よりも大きい場合、マルチキャストビットを設定するステップから成ることを特徴とする請求項15に記載の方法。

【請求項18】 上記間接ビットが1に等しくなく、またタイプビットがノードに等しくない場合に結果を保存するステップはさらに、上記間接ビットが1に等しい場合、あるいは上記タイプビットがノードに等しい場合、エラーコンディションを示すステップから成ることを特徴とする請求項15に記載の方法。

【請求項19】 ルートルックアップを実行するステップはユニキャストルックアップを実行することから成ることを特徴とする請求項15に記載の方法。

【請求項20】 ルートルックアップを実行するステップはマルチキャストルックアップを実行することから成ることを特徴とする請求項15に記載の方法。

【請求項21】 ルートルックアップリクエストを受取るステップと、

ルートルックアップを実行するステップであって、そのルックアップが該リクエストに対する最初のルックアップであるかの判断を行うステップと、

それが該リクエストに対する最初のルックアップである時、該リクエストに対し現ブロックデータを使用し、それが該リクエストに対して最初のルックアップでない

時、前回ブロックデータを使用するステップと、該ブロックデータが葉を示しているかどうかの判断を行うステップであって、

該ブロックデータが葉を示している時、間接ビットが該ブロックデータに設定されているかどうかを判断し、該間接ビットが設定されている時、該ブロックデータからのルートテーブルルックアップエンジンルートから成るルートテーブルルックアップアドレスを供給し、該ルックアップテーブルから読取りを行い、該読取りは新規ブロックデータを供給し、該間接ビットが1に等しくないか、またタイプビットがノードに等しくないかの結果を保存するこれらステップを実行するステップと；該間接ビットが設定されていない時、結果を保存するステップを実行するステップとから成ることを特徴とするルートルックアップエンジンの動作方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 データ転送においてルートルックアップを実行するルートルックアップエンジンに関

する。

【0002】

【従来の技術】 ルータのようなネットワークデバイスは処理動作のその一部としてルートルックアップを実行しなくてはならない。ルートルックアップは、次にパケットがどこに送られるべきかを判断するために、ルータにより受け取られる各パケットごとに必要となる。受信パケットは32ビット長の宛先アドレスを含む。宛先アドレスはパケットの転送がなされる固有のアドレスを識別する。ルートルックアップでは受信パケットの宛先アドレスを使用して、ルーティングテーブルにおける次ホップアドレスと出口ポートを識別する。宛先アドレスが32ビット長である場合、宛先可能数は10億以上となる。インターネットプロトコル(IP)アドレスはネットワーク番号とホスト番号から構成されている。ネットワーク番号は可変長であり、固有のネットワークを識別する。ネットワーク番号に連結しているのがホスト番号であり、ホスト番号は特定のネットワーク内の固有のホストシステムを識別する。受信アドレスのネットワーク番号部分はルーティングテーブルにおける複数エントリーとマッチングが行われる。この最適マッチは、アドレスの最初の部分のビットが最大数マッチングするものである。

【0003】

【発明が解決しようとする課題】 しかしながら、この従来のルートルックアップにおいてマッチングを実行するために使用される従来技術による方法は複雑、かつ時間がかかる。よって、ルートルックアップを迅速、かつ効果的に実行するルートルックアップエンジンが必要とされる。

【0004】

【課題を解決するための手段】 上述した問題を解決するため、本発明は次ホップインデックスを判断するルートルックアップエンジンを開示する。ルートルックアップエンジンはルックアップキーを受け取り、プレフィックスエキスパンションと可変ストライド(stride)ツリーキャプチャによりマルチビット樹木探索を実行する。ルートルックアップエンジンが返送するデータは、次ホップ情報およびステータスフラグより構成される。ルートルックアップエンジンはコンパクトでフィールド再利用可能なデータ構造を使用する。ルートルックアップエンジンでは仮想プライベートネットワーク上にユニキャストIPアドレスルックアップおよびマルチキャストIPアドレスルックアップの両方を実行する。ルートルックアップエンジンは別々のインデックスメモリーと転送メモリーを使用する。ルートルックアップエンジンのサーチタイムの上界はルートテーブルサイズに関係なく決定される。

【0005】

【発明の実施の形態】 ルートルックアップエンジンを開

示す。ルートルックアップエンジンは多数アプリケーションにおいて使用が可能である。望ましい実施形態においては、ルートルックアップエンジンは非同期トランスファモード(ATM)スイッチのインターネットプロトコル(IP)ルーティングサブシステムの一部である。ATMスイッチのルーティングサブシステムは迅速かつ効果的なルートルックアップメカニズムを提供する。図1はATMスイッチのトップレベルブロック図である。ATMスイッチは入/出力アダプタ52を備えている。この入/出力アダプタはインターフェースオブジェクト、同期光ネットワーク(SONET)フレーム、およびハイレベルデータリンク制御手順(HDLC)コントローラを備えている。ATMスイッチはまた、入/出力モジュール10を備えている。入/出力モジュールは複数の転送エンジン20を備えている。さらに入/出力モジュールはタイミング制御ユニット30、DC電流変換/分配ユニット40、スイッチファブリックインターフェース50、および制御プロセッサ60を備えている。入/出力モジュールのタイミング制御ユニット30はローカルタイミングのシステムリファレンスを物理層に提供する。入/出力モジュールのDC電流変換/分配ユニット40は直流直流コンバータより成る。バックプレーインより可能な48ボルトはモジュールに必要な電圧に変換される。

【0006】スイッチファブリックインターフェース50は、スイッチファブリック経由で各転送エンジンとスイッチにおける他のカードとを連結する。データはATMセル形式で搬送される。スイッチファブリック過密情報もまたファブリックから返送され、転送エンジン用に再フォーマット/サマライズされる。

【0007】制御プロセッサ60は、バックグラウンド汎用プロセッサとしてベンディウム(登録商標)マイクロプロセッサにより構成される。転送エンジン20については以下に詳細に記載する。

【0008】図2は入/出力モジュールの単一転送エンジン20のブロック図である。転送エンジン20は、制御メモリ160およびパケットメモリ170と連動するインバウンドセグメンテーション/再アセンブリ(SAR)デバイス150、制御メモリ190およびパケットメモリ200と連動するアウトバウンドセグメンテーション/再アセンブリ(SAR)180、パケットヘッダディストリビュータ210、パケット分類エンジン110、ルートルックアップエンジン100、転送プロセッサ130、転送プロセッサと連動するキャッシュ140、およびDRAM/SRAM120から構成されている。

【0009】転送エンジン20はインバウンドパケットおよびアウトバウンドパケットを処理する。インバウンドパケットに関して、パケットはインバウンドSAR150にトランスファされ、そこでパケットはパケットメ

モリ170に保存される。インバウンドSAR150は最高サービス品質パケットをパケットヘッダディストリビュータ210に送る。転送プロセッサ130はパケットヘッダディストリビュータのステータスチェックを行い、インバウンドパケットの受取りがいつ可能になるかを判断する。転送プロセッサ130は、パケットヘッダディストリビュータFIFOからのプライオリティの最も高いヘッダーを検索し、これをキャッシュ140に保存する。転送プロセッサ130はヘッダーを処理し、ルートルックアップエンジン100とパケット分類エンジン110の両方に要求されたヘッダーエレメントを同時に送信する。そこで、それらはイグレス(egress)キューに保存される。ルートルックアップエンジン100はパブリックあるいは仮想プライベートネットワークユニキャストルックアップを実行し、必要な場合、次のマルチキャストルックアップを実行する。一方で、パケット分類エンジン110はヘッダーエレメントに対してフィルタチェックを実行する。

【0010】ルートルックアップエンジン100とパケット分類エンジン110のパケットヘッダディストリビュータ210へのステータスパスが完了すると、転送プロセッサ130によりボーリングされるイグレスキューに結果を保存する。ルートルックアップエンジン100の動作については詳細を後述する。転送プロセッサ130はボーリングを行い、次に、新規ヘッダーが公式化されるルートルックアップと分類結果を得る。次に伝送情報に加えて新規ヘッダーがパケットヘッダディストリビュータ210に書込まれる。インバウンドSAR150は新規にフォーマット化されたヘッダーを検索し、それをパケットの残りとともにパケットメモリ170に返送して、そして、セグメント化されスイッチファブリックに伝送されるパケットをスケジュールする。

【0011】転送エンジン20はアウトバウンドパケットを次のように処理する。ATMセルはスイッチパケットからアウトバウンドSAR180にトランスファされ、そこで全体パケットが再アセンブリされ、パケットメモリ200に保存される。アウトバウンドSAR180は最高サービス品質パケットのヘッダーをパケットヘッダディストリビュータ210に送信する。転送プロセッサ130はパケットヘッダディストリビュータ210のステータスチェックを行い、アウトバウンドパケットの受け取りが可能であることを判断する。転送プロセッサ130はパケットヘッダディストリビュータFIFOからプライオリティの最も高いヘッダーを検索して、それをそれらのキャッシュに保存する。転送プロセッサ130はヘッダーを処理し、ルートルックアップエンジン100とパケット分類エンジン110の両方に要求されたヘッダーエレメントを同時に送信する。そこでそれらはイグレスキューに保存される。ルートルックアップエンジン100はパブリックあるいは仮想プライベート

ートネットワークユニキャストルックアップを実行し、必要な場合、次のマルチキャストルックアップを実行する。一方で、パケット分類エンジン110はヘッダーエレメントに対してフィルタチェックを行う。

【0012】ルートルックアップエンジン100とパケット分類エンジン110のパケットヘッダーディストリビュータ210へのステータスパスが完了すると、転送プロセッサ130によりポーリングが行われるイグレスキューとなる。転送プロセッサ130はポーリングを行い、次に、新規ヘッダーが公式化されるルートルックアップと分類結果を得る。次に伝送情報に加えて新規ヘッダーがパケットヘッダーディストリビュータ210に書込まれる。アウトバウンドSAR180は新規にフォーマット化されたヘッダーを検索して、パケットの残りとともにパケットメモリ200にそれを返送し、物理フレームインターフェースに伝送されるパケットをスケジュールする。

【0013】図3は転送エンジン20のルートルックアップエンジン100のブロック図である。ルートルックアップエンジンは転送プロセッサ130の次ホップインデックスを判断するハードウェアサーチエンジンである。ルートルックアップエンジン100は、転送プロセッサインターフェース65、入/出力FIFO40および入/出力FIFO50、ルックアップコントローラ20、ルックアップテーブルDRAM30、および制御プロセッサインターフェース70とにより構成される。転送プロセッサインターフェース65はサーチエンジンのキーをロードするために使用される。キーは、物理ポート/論理ポートに固有のツリー根のアドレス/仮想プライベートネットワークID、IP宛先アドレス、IPソースアドレス、ツリーの第一レベルのオーダ、および、シーケンスエラーチェック用にヘッダーといっしょに返送されるオプションリクエストIDとにより構成される。制御プロセッサインターフェース70はルックアップ間のテーブル更新に使用される。

【0014】その動作において、転送プロセッサは、ルートルックアップエンジンとパケット分類エンジン両方の共通アドレスに1つあるいはそれ以上のキャッシュラインを同時に書込む。転送プロセッサインターフェース65はルートルックアップエンジンとパケット分類エンジン両方からの結果を同時に検索するために使用される。これらは、パケット分類エンジンより供給される結果部分とルートルックアップエンジンより供給される結果部分と共に同一アドレスから同時にレポートする。

【0015】ルートルックアップエンジンはFIFOもしくは他のインターフェースを備え、スループットを最大限にすることを可能にする。入/出力FIFOは、転送プロセッサからの複数ルックアップリクエストを所定時間で明確にする。これらのパケットが異なるルックアップ時間を有する場合、FIFOはいくつかのヘッダー

バーストをイン/アウトさせることによりルックアップレートを平滑にする。ルートルックアップエンジンは仮想プライベートネットワーク(VPN)上でユニキャストIPアドレスルックアップおよびマルチキャストIPアドレスルックアップの両方を実行する。各仮想プライベートネットワークは、その仮想プライベートネットワークに特定のテーブル根に対するポインタとして使用される。多重仮想プライベートネットワークはソフトウェアの間接処理により同一ルートテーブルをマップする。

【0016】IPパケットを転送するために、パワープロセッサコントローラはルートルックアップリクエストをパケットヘッダディストリビュータに提示する。ルートルックアップエンジンのハードウェア可変ストライドツリーテーブルは、ルートルックアップエンジンからの回答と共に含まれる結果インデックスによりリクエストを解決するために使用される。インデックスがパワープロセッサコントローラに対し使用可能となると、これらは転送テーブルからの転送情報への直接アクセスのために使用される。

【0017】ルートルックアップからの結果は、次ホップインデックス、有効ルックアップ/ルート未発見ビット、ステータスフラグ、マルチキャストルックアップ実行ビット、統計情報、および、シーケンスエラーチェックのためのオプションリクエストIDにより構成される。

【0018】パケットがマルチキャストパケットである場合、ルートルックアップエンジンサーチはまず最初にIP宛先アドレスにおいて実行され、次にIPソースアドレスにおいて実行される。サーチにはブレフィックスエクスパンションおよびキャプチャによるマルチビット樹木探索を用いる。次ホップインデックスが発見されたかあるいはキー終了に達した場合にサーチは終了となる。この試みにおいてIPソースアドレスの終了時にIPヘッダーサーチの64ビット超を継続させると、ハードウェアエラーチェックはサーチを終了させ、エラーリポートを行う。ルーティングサブシステムのルートルックアップエンジンマネジャーはルートルックアップエンジンメモリ管理を行う。ルートルックアップエンジンメモリは、デフォルト仮想プライベートネットワーク(VPN)、VPN0を含む、構成された仮想プライベートネットワーク(VPN)各々に対するハードウェア可変ストライドツリー(VST)ルートテーブルを保存するために使用される。各ハードウェアVSTは最小の近接メモリスペースが保証されている。残りのメモリスペースはメモリブロックのフリーブールとして利用可能である。ルートがVSTテーブルに追加されると、また、これから削除されると、メモリブロックがこれらメモリブールから割当てされ、そして、これらメモリブールに解除される。ルーティングサブシステムはメモリの断片化を最小に保つよう管理する。

【0019】ルーティングサブシステムは、パワープロセッサコントローラのメモリ上の転送データベースレジデントとルートルックアップエンジンのメモリ上のハードウェアVSTテーブルレジデントとを確実に同期させておく。同様に、カーネルプロセッサのメモリ上のソフトウェアVSTテーブルレジデントはルートルックアップエンジンのハードウェアVSTテーブルと同期を保たなくてはならない。

【0020】ルートルックアップエンジンはローカルメモリに備わったVSTベースルートルックアップテーブルを含む。これらのVSTテーブルは1つの特定ルートに一致する。ハードウェアVSTテーブルの各エントリは、パワープロセッサコントローラメモリ上の転送テーブルレジデント内における相当転送エントリに対するインデックスを有する。

【0021】各VPNに関して、シングルソフトウェア*

*VSTはカーネルプロセッサのメモリにて維持され、相当のハードウェアVSTはルートルックアップエンジンメモリにて維持される。VSTルートテーブルは高レートにて最適マッチングプレフィックスをサポートするVSTデータ構造にて構築される。VSTテーブルはサーチキーを別個のビットストリングに分割することで構築される。サーチキーはIPアドレスである。VSTレベル数はルートルックアップ中の最大メモリアクセス数を特定する。このように、ルックアップを完了する時間はバウンドされるが決定性を持つものではない。各VSTレベル長はスライド長として知られ、その区分を表すノード内のエレメント数を判断する。

【0022】ハードウェアVSTデータ構造は次のようになる。

エレメント

【表1】

31	30	29	28 24	23 0
タイプ	インダイレクト	マルチキャスト	オーダ	ポインタ

タイプ : 0=葉, 1=ノード

オーダ : ノードについてlog₂、葉についてプレフ

ィックス長

ポインタ : 次ノードのベースアドレスに対するポイン

20×ンタ(無効の場合ゼロ、または転送エントリに対するインデックス(無効の場合0 x F F F F F F))

ノード

【表2】

エレメント[ビットストリング値=0]
エレメント[ビットストリング値=1]
.....
エレメント[ビットストリング値=2 ^{node order} -1]

【0023】ハードウェアルートサーチデータ構造はノードにより構成され、その各々はエレメントのアーレイである。ノードにおけるエレメント数はノードオーダにより決定する。ここでオーダは、このノードの評価に使用されるサーチキー(宛先IPアドレス)のビット数である。各エレメントは、葉か他のノード根のどちらかとして次ノードを指示するタイプインジケータを有する。現在のノードのタイプが「ノード」である場合、オーダは次ノードのオーダを示す。また、現在のノードのタイプが「葉」である場合、オーダは葉のプレフィックス長を★

★示す。ポインタフィールドはエレメントタイプにより決定する。タイプが「葉」である場合、ポインタはフォアグラウンドの転送テーブルのインデックスであり、宛先アドレス/マスクペアの転送エントリを識別する。タイプが「ノード」である場合、ポインタは評価される次ノードのアドレスを識別するサーチスペースのポインタである。次のテーブルはオーダの意味と異なるタイプのポインタフィールドを示したものである。

【表3】

タイプ	オーダ	ポインタ
0(ノード)	Log ₂ (ポインタにてノードにおけるエレメント数)	ノードに対するポインタ
1(葉)	エントリポインタの(プレフィックス長-1)	転送エントリに対するインデックス

【表4】

ルートルックアップエンジンテーブルエントリ:

ビット	31	30	29	28 24	23 0
フィールド	P	T	I(1)	オーダ(5)	RLE-Root/Flag, NHP
	(1)	(1)			

P: パリティビット。奇数パリティはハードウェアによりオプションでSDRAMに書込まれ、SDRAMより読取られる。{1ビット}

T: タイプビット 0=ノード, 1=葉 {1ビット}

I: 間接ビット。1=間接, 0=ダイレクト {1ビット}

オーダ: 1から32に相当する次レベルのオーダ {5ビット}

RLE-Root/Flag.NHP: (ルートルックアップエンジン(RLE)根/フラグ、次ホップポインタ): Tビットによってこのデータフィールドに対し2コンテキストが存在する。ノードの場合、次ノードポインタ {24ビット}、または葉の場合、RLE次ホップポインタ {16ビット} にプリペンドされたRLEフラグ {8ビット}。

【0024】ルートルックアップエンジン根はサーチ開始に使用される。ルートルックアップエンジン根は根ポインタとツリーの第一レベルのオーダを含む。第一ルックアップアドレスは根ポインタ、プラス、インデックスにより構成される。インデックスは宛先アドレスの上位ビットから導出される。ビット数は可変であり、第一レベルのオーダによって示される。このアドレスはテーブルからのエレメントをルックアップするために使用される。このエレメントが葉である場合、サーチは完了し、エレメントはバケット分類エンジン/ルートルックアップエンジンマネジャーにもどされる。エレメントがノードである場合、ポインタは次のルックアップに使用される。

【0025】次ルックアップのアドレスは上記のポインタ、プラス、インデックスから成る。インデックスは宛先アドレスからの「次」ビットから導出される。正確なビットは、最終レベルのオーダ、プラス、このレベルのオーダから導出される。ゆえに、インデックスに使用される宛先アドレスの相当ビットを知る目的で、各レベルのオーダがトラックされなくてはならない。

【0026】このアドレスはテーブルのエレメントをルックアップするために再度使用される。このエレメントが葉である場合、サーチは完了し、このエレメントはバケット分類エンジン/ルートルックアップエンジンマネジャーにもどされる。このエレメントがノードである場合、ポインタは次のルックアップに使用される。これは葉が見つけ出されるまで繰り返される。アクセス数やレベル数が最大数を超過した場合には、代わりにエラーが返送される。

【0027】ある構成においてはルートルックアップに間接方法を有することが望ましい。これは間接ビットを用い達成される。間接ビットが設定されると、葉フラグは無視され、オーダフィールドは使用されず、そのインデックスロケーションから返送されたルート結果により、現24ビットポインタがダイレクトテーブルインデ

ックスとして使用される。間接ファンクションはデブスをゼロに設定することにより呼び出すことも可能である。

【0028】図4は、本発明によりルートルックアップを実行する方法300のフローチャートである。ルックアップはステップ310にてリクエストを受け取ることから開始する。ステップ320において、これがそのリクエストに対する最初のルックアップであるかどうかの判断が行われる。これがリクエストに対する最初のルックアップでない場合、ステップ330に示すように、TYPE、INDIRECT、ORDER、およびRLE ROOTの前回値が使用される。これがリクエストに対する最初のルックアップである場合、TYPE、INDIRECT、ORDER、およびRLE ROOTがリクエストブロックデータから取り出される。

【0029】ステップ350において、TYPEビットが評価される。タイプが「葉」でない場合、ステップ420からステップ480が実行される。ステップ420において、DRAMアドレスはRLE ROOT、プラス、ゼロ、これに連続する宛先アドレスとソースアドレスに設定される。

【0030】次にステップ430において、DRAMアドレスからの読取りが実行され、TYPE、INDIRECT、ORDER、およびRLE ROOTの新規値が得られる。ステップ440において、現在のオーダが、現オーダ、プラス、オーダに設定される。

【0031】ステップ450において、現在のオーダが64よりも小さいかあるいは等しいかの判断がなされる。そうでない場合、エラーコンディションが示されてサーチは終了する。現在のオーダが64よりも小さいかあるいは等しい場合には、現在のオーダが32より大きいかがステップ470にて判断される。もしそうでない場合、ステップ320より開始する他のルックアップが実行される。現在のオーダが32により大きい場合、ステップ480が実行され、マルチキャストオペレーションを示すLビットが設定される。ステップ480の後、ステップ320より開始する他のルックアップが実行される。

【0032】ステップ350に戻り、TYPEビットが「葉」でなかった場合、ステップ360にて間接ビットが設定されるかどうか判断される。間接ビットが設定されない場合、ステップ410で示すようにルックアップは完了し、その結果が保存される。

【0033】間接ビットがステップ360にて設定されると、ステップ370において、DRAMアドレスがRLE ROOTに設定される。ステップ380において、RLE DRAMのアドレスが読取られ、TYPE、INDIRECT、ORDER、およびRLE ROOTの新規値が得られる。ステップ390において、INDIRECTビットが設定される場合、あるいはタ

10

20

30

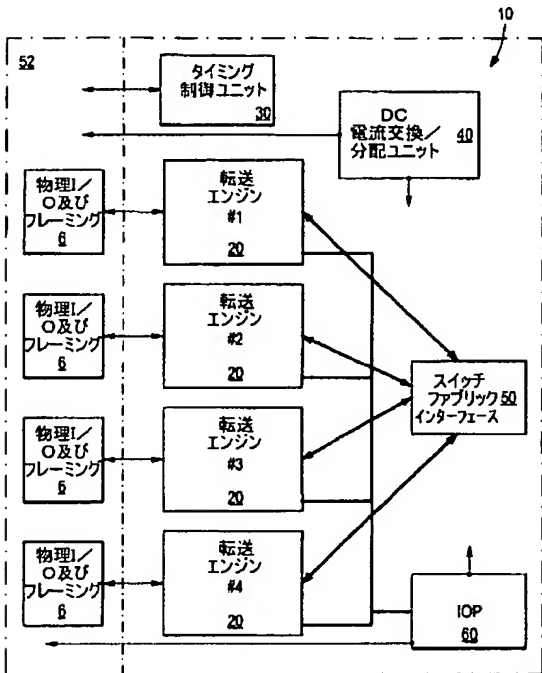
40

50

イブが「ノード」の場合、エラーコンディションが示され、ステップ400で示すようにルックアップは終了する。INDIRECTビットが設定されないか、あるいはTYPEビットがノードを示さないかのどちらかである場合、ステップ410で示すようにルックアップは完了となり、その結果が保存される。

【0034】以上、本発明の実施形態を詳細に説明したが、本発明の範囲を逸脱することなく他の方法でも具体化出来ることは当業者にとって明らかである。 *

【図1】



*【図面の簡単な説明】

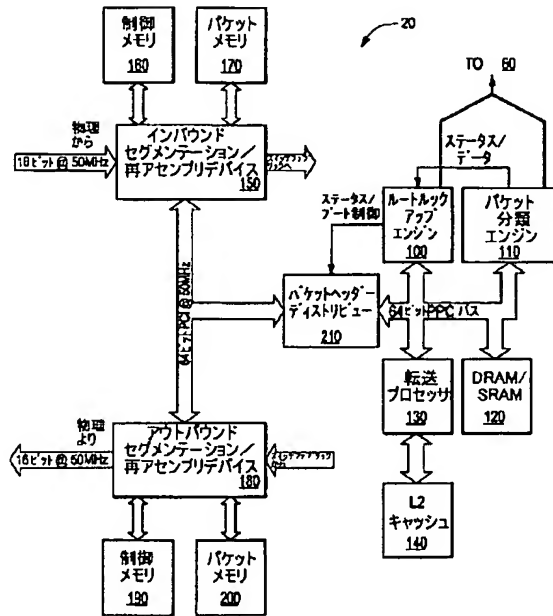
【図1】 ATMスイッチのトップレベルブロック図である。

【図2】 転送エンジンのブロック図である。

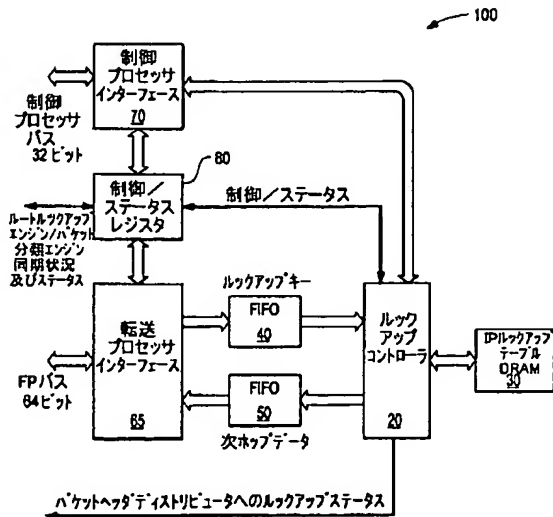
【図3】 本発明によるルートルックアップエンジンのブロック図である。

【図4】 本発明に基づいてルートルックアップを実行するフローチャートである。

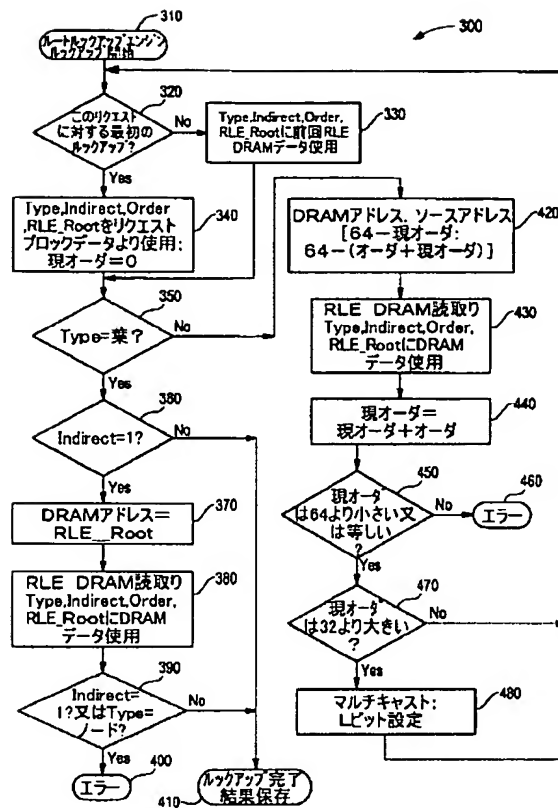
【図2】



【図3】



【図4】



フロントページの続き

(72)発明者 サンジェイ ジー. チェリアン
 アメリカ合衆国 03033 ニューハンプシ
 ャー, ブルックリン, マックスウェル ド
 ライヴ 6

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.